

# How to build (good) models

Andreas Handel

University of Georgia

2019-07-10

# Modeling workflow overview

- Have a scenario/question for which a model can provide useful insight (hardest part).
- Translate your scenario/question into a suitable model (hard).
- Implement the model on a computer, analyze it, produce results such as figures and tables (fairly easy).
- Write it all up and try to publish (easy to hard, depends on journal).

# Step 1 - Should I?

- What do I want to accomplish and can a model help?



[dilbert.com](http://dilbert.com)

**Coming up with an important/relevant question where a model can be useful is the most important component.**

# Step 2 - Decide what matters

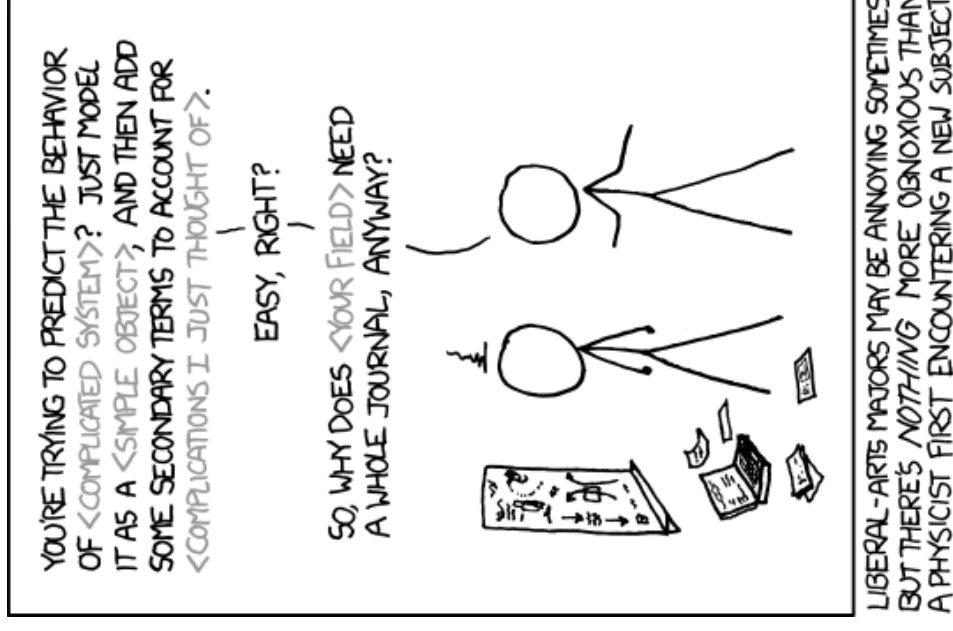
- Reality is so complicated, we can never (and don't want to) include every detail into a model. Therefore, models are always approximations of the real world.
- The type of model and detail needed depends on the question.
- Good modeling means having a model that includes details that matter and ignores details that do not matter.



Maps as an analogy for models.

# Step 3 - Choose model type & details

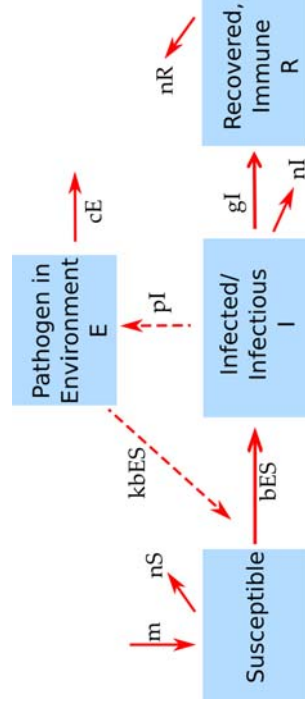
- I recommend starting with a simple model.
- ODE models are good starting points.
- Only include the most essential components and processes you know you need.
- You can always increase model size/complexity later.



xkcd.com

# Step 4 - Sketch the model

- Draw diagrams of the model components and interactions.
- If you do an ODE model, write down the equations.
- Writing down the ODEs might be a good exercise even if you will use a more complicated model eventually.

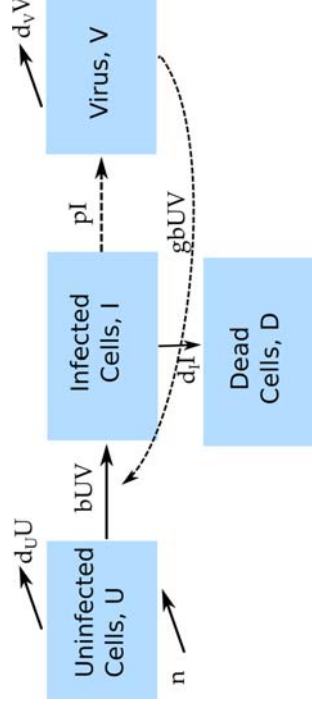


$$\dot{S} = m - nS - bSE$$

$$\dot{I} = bSE - gI - nI$$

$$\dot{R} = gI - nR$$

$$\dot{E} = pI - cE - kbSE$$



$$\dot{U} = m - d_U U - bUV$$

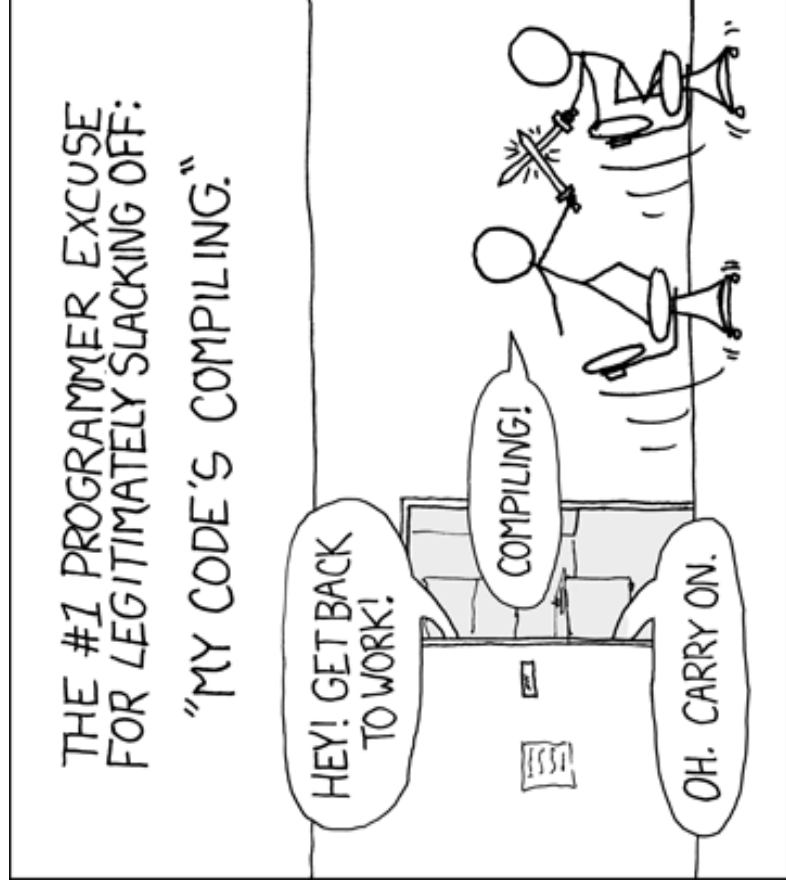
$$\dot{I} = bUV - d_I I - nI$$

$$\dot{D} = d_I I$$

$$\dot{V} = pI - d_V V - gbUV$$

# Step 5 - Implement the model

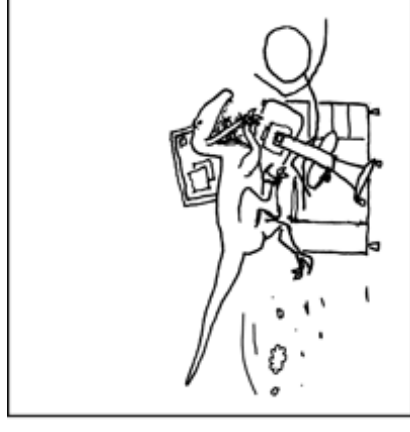
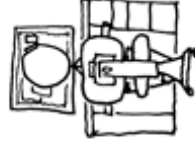
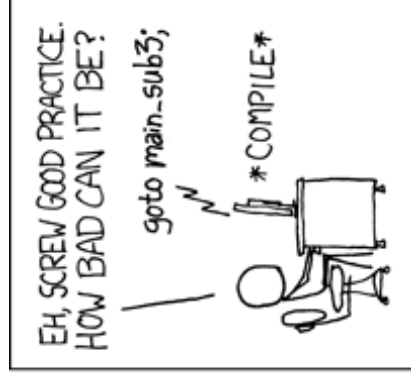
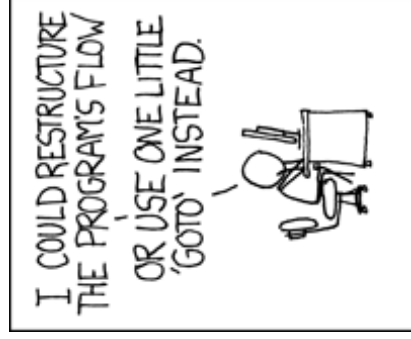
- Write computer code for the model.



xkcd.com

# Step 6-666 - debug code, fix model

- Run your model code. It will likely not run and/or produce weird results.
- Find all the bugs in your code and in your conceptual/mathematical model until everything seems to work.
- As you test your model, make sure every component works right.
- Trying extreme scenarios is often a good quick test.



xkcd.com



# Step 667 - analyze model

- Once your model works, run simulations, analyze results.



[dilbert.com](http://dilbert.com)

# Step 668-N - Reality check & model adjustment

- Compare your model results with whatever data you have/know about.
- As needed (very likely) adjust model to reality.
- **Note the model "failures"**: If you built a reasonable model and it wasn't able to reproduce reality, you/we have learned something.
- Keep iterating until the model "works".



# Step N+1 - Write up and publish

## Your (real) Impact Factor

$$\text{Impact Factor (corrected)} = \frac{\begin{array}{l} \text{\# times your work is cited} \\ - \text{\# citations that actually trash your work} \\ + \text{\# times you were cited just to pad the introduction section} \\ - \text{\# times you cited yourself (nice try)} \end{array}}{\begin{array}{l} \text{\# original articles you've written} \\ + \text{\# articles you were included in out of pity or politics} \\ + \text{\# not-so-original articles you've} \\ \text{\textit{written}} \\ \text{copied and pasted} \end{array}} \quad \begin{array}{l} \text{\# citations the editor pressured the author to include to increase the journal's impact factor} \end{array}$$

JORGE CHAM © 2008  
[WWW.PHDCOMICS.COM](http://WWW.PHDCOMICS.COM)